REMARKS

As a preliminary matter, Applicant appreciate Examiner Steelman's courtesy in granting an interview to discuss the application. Though no agreement was reached, Applicant believe significant progress was made toward an understanding of the claimed subject matter.

The Examiner states that Applicant's previous amendment to the specification at page 8, line 4 was unclear. In response, Applicant submits that the line was amended to remove the dash in the middle of "phase behavior".

Claim 12 and dependent claims 13-21 & 26 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. Applicant is unclear as to whether claims 41, 42, and 47, and their respective dependent claims, remain rejected, as the Office Action also states that the 35 U.S.C. § 112 rejection of these claims has been withdrawn. In response, Applicant has amended claim 12 as suggested by the Examiner, and submits that the rejection has been overcome.

Claims 1-17, 22-33, 35-40, 42-48, 50, and 52 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Goodnow. Applicant respectfully traverses the rejection, as Goodnow fails to teach or suggest at least comparing at least one identified behavior for at least one interval of execution to another interval of execution to determine similarity between the intervals of execution, as defined in independent claim 1, identifying behavior of a hardware-independent metric within at least one arbitrary section of execution of a portion of a computer program and classifying each of the at least one arbitrary section of execution according to the identified behavior into clusters of similar behavior, as defined in independent claim 27, and identifying behavior of a hardware-independent metric for each of a plurality of intervals of execution and comparing the identified behavior of each of the plurality of intervals to an identified target behavior to determine a representative interval, as defined in independent claim 35.

As previously submitted by Applicant, Goodnow fails to teach analyzing intervals of execution or arbitrary sections of execution at all, but instead specifically teaches analyzing code segments. The present specification defines "interval" as a selection of continuous instructions in program execution order. "Arbitrary section of execution", has a similar meaning. However, as would be understood by reference to the present specification, this interval or arbitrary section is not set by a particular part of the underlying code, but instead it is independent of the particular code being executed. As a result, certain embodiments of the present invention can identify the code that actually is being run as a tracked behavior of a particular interval or arbitrary section.

In other words, within a particular arbitrary section or interval of execution, continuous instructions are executed, regardless of which instructions are executed, until a particular point, such as when a number of instructions has occurred, an amount of time has passed, a particular metric has been reached, etc. In addition to the meaning of "intervals of execution" and "arbitrary section of execution" as would be understood by a person of ordinary skill in the art having reference to the present specification, Applicant has amended dependent claims 9 and 10 to more specifically define at least these differences.

The Office Action relies on the "instruction interval" definition to maintain a rejection based on Goodnow, and thus equates a code segment with an instruction interval. However, an instruction interval refers to a number of continuous instructions that are executed, regardless of which specific code (e.g., code language) is executed. Goodnow always compares or analyzes specific code segments - that is, segments necessarily fixed in and dependent on the language of the code itself, such as the code segments defined in Table 1 at column 5. These code segments can only be the parts of code that fall within the particular lines defined.

Goodnow does not teach or remotely suggest comparing intervals or sections of execution that are independent of the code. The purpose of Goodnow's invention is to analyze a block of code (that is, the specific instructions spelled out in specific lines of code) to determine its effect before incorporating it into another program. This would not work if intervals of execution or arbitrary sections of execution were analyzed or compared (e.g., a programmer would likely not take an arbitrary set of continuous executed instructions, regardless of what code was actually run, and copy it into another program).

For at least these reasons, Applicant respectfully submits that claims 1, 27, and 35 and their respective dependent claims are allowable over the references of record, including Goodnow. The 35 U.S.C. § 103 rejection is traversed for at least the reasons stated above regarding the deficiencies of Goodnow regarding claims 1, 27, and 35, and for at least the reason that the secondary references cited fail to remedy the deficiencies of Goodnow.

For at least the foregoing reasons, Applicant believes that this case is in condition for allowance, which is respectfully requested. The Examiner should call Applicant's attorney if an interview would expedite prosecution.

Respectfully submitted,
GREER, BURNS & CRAIN, LTD.

By:
Arik B. Ranson
Registration No. 43,874

Customer No. 24978
June 29, 2007
300 South Wacker Drive
Suite 2500
Chicago, Illinois 60606
Telephone: (312) 360-0080
Facsimile: (312) 360-9315
P:\DOCS\0321\68199\BO8025.DOC